



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/668,124	09/22/2003	Sheng Liang	188135/US/3	2001
66083	7590	10/05/2009		
DORSEY & Whitney LLP on behalf of Sun Microsystems, Inc. 370 SEVENTEENTH ST. SUITE 4700 DENVER, CO 80202-5647			EXAMINER HOANG, PHUONG N	
			ART UNIT 2194	PAPER NUMBER
			MAIL DATE 10/05/2009	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/668,124	Applicant(s) LIANG ET AL.	
	Examiner PHUONG N. HOANG	Art Unit 2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 26 June 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 14 - 17 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 14 - 17 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 14 – 17 are pending for examination. This office action is in response to amendment filed 6/26/2009.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. **As to claims 14 - 17**, the claim “computer readable storage medium” in claims 14 – 17 are non-statutory. In view of Applicant’s disclosure, paragraph 0036, the computer readable storage medium is not limited to storage media embodiments, instead being defined as including both storage media embodiments (e.g., CD-ROM, disk, magnetic tapes) and transmission media embodiments (e.g., data signal, carrier wave). As such, the claim is not limited to statutory subject matter and is therefore non-statutory. To overcome this type of 101 rejection, the claims need to be amended to include only the physical computer media (e.g., storage media such as floppy disk, system memory, and hard drive) and not a transmission media or other intangible or non-functional media. Or, the transmission media or other intangible or non-functional media in the specification can be deleted so that the media only comprise floppy disk, system memory, or the like.

Claim Rejections - 35 USC § 103

Art Unit: 2194

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. **Claims 14 – 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Folliot, “A Dynamically Configurable, Multi-Language Execution Platform” pages 175 – 181, in view of Javasoft “Java Native Interface Specification” pages 1 - 82.**

6. Folliot reference was cited in previous office action.

7. **As to claim 14**, Folliot teaches an apparatus for providing a modular runtime environment architecture comprising:

a single virtual machine implementation (Virtual virtual machine (VVM), section 1 page 176 paragraph 4 – section 2), embodied in at least one tangible computer readable storage medium (memory, page 177) and executable utilizing at least one processor (virtual processor, page 178), the single virtual machine providing virtual machine functionality (virtual virtual machine can be configured to execute applications, section 2 pages 176 – 177);

a plurality of support libraries, each support library of a plurality support libraries includes library functions (VMlets, each VMlet include five things..... VP instructions, section 2 pages 177 - 178), the virtual machine functionality supported by each support

Art Unit: 2194

library of the plurality of support libraries (VMM is the capability to describe virtual machines (called VMlet) and to associate one VMlet with each application, VMM is transparent loading and unloading of these VMlets at runtime, section 2 pages 167 – 177);

virtual machine capable of replacing the first support library, and the virtual machine implementation without modification of the virtual machine implementation (unload and load the VMlets, section 1 page 176 - section 2).

Folliot does not explicitly teach the step of the virtual machine provides a bi-directional virtual machine interface for communication between a first support library and the single virtual machine implementation, the bi-directional virtual machine interface defining virtual machine implementation-dependent operations performed by the virtual machine implementation, and the bi-directional virtual machine interface supports communication between a second support library from the plurality of support libraries.

JavaSoft teaches the virtual machine provides a bi-directional virtual machine interface for communication between a first support library and the single virtual machine implementation, the bi-directional virtual machine interface defining virtual machine implementation-dependent operations performed by the virtual machine implementation, and the bi-directional virtual machine interface supports communication between a second support library from the plurality of support libraries (the Java Native Interface (JNI) is a native programming interface. It allows Java code that runs inside a

Java Virtual Machine (VM) to interoperate or bi-directional interface with application and libraries, chapter 1, especially page 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Folliot and Javasoft because the JNI provides no restrictions on the implementation of the Java virtual machine that allows Java vendors can add support for the JNI without affecting other parts of the virtual machine (Javasoft, page 1).

8. **As to claim 15**, Folliot teaches an article providing a runtime environment, embodied in at least one tangible computer-readable storage medium (memory, page 177), comprising:

a virtual machine implementation, embodied in at least one tangible computer-readable storage medium (memory, page 177) and executable utilizing at least one processor (virtual processor, page 178), that uses a first support library implementation, the first support library implementation can be replaced by a second support library implementation without modifying the virtual mechanism implementation (unload and load the VMlets, and there is no need modification, section 1 page 176 - section 2).

Folliot does not explicitly teach the step of a bi-directional virtual machine interface enables communication between the virtual machine implementation and the

Art Unit: 2194

first support library and communication between the virtual machine implementation and the second support library.

JavaSoft teaches a bi-directional virtual machine interface enables communication between the virtual machine implementation and the first support library and communication between the virtual machine implementation and the second support library (the Java Native Interface (JNI) is a native programming interface. It allows Java code that runs inside a Java Virtual Machine (VM) to interoperate or bi-directional interface with application and libraries, chapter 1, especially page 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Folliot and JavaSoft because the JNI provides no restrictions on the implementation of the Java virtual machine that allows Java vendors can add support for the JNI without affecting other parts of the virtual machine (JavaSoft, page 1).

9. **As to claim 16**, Folliot teaches a virtual machine implementation (VVM, section 1 page 176 – section 2), embodied in at least one tangible computer-readable storage medium (memory, page 177) and executable utilizing at least one processor (virtual processor, page 178), suitable for use in a runtime environment, the virtual machine implementation comprising:

the virtual machine implementation suitable for use in conjunction with a plurality of support libraries, including the first support (Support for each specific language within a family is provided by a VMlet, each VMlet include five things..... VP instructions,

Art Unit: 2194

section VMlets on page 178), the first support library can be replaced by a second support library without altering the virtual machine implementation (unload and load the VMlets, and there is no need modification, section 1 page 176 paragraph 4 - section 2);

at least some library functions that call virtual machine implementation-dependent functions that perform operations that are dependent on the particular virtual machine implementation used to perform the operations (each VMlet includes five things: mapping from the VM bytecodes to the VVM's execution mechanism, section VMlets on page 178);

run time environment (execution/run time environment, page 176 paragraph 4 and section 2).

Folliot does not explicitly teach:

a virtual machine interface defining a number of operations performed by the virtual machine implementation;

the virtual machine interface defines operations performed by the virtual machine implementation-dependent;

the virtual machine interface supports communication between a first support library and the virtual machine implementation.

Javasoft teaches:

a virtual machine interface defining a number of operations performed by the virtual machine implementation (the Java Native Interface (JNI) is a native programming interface. It allows Java code that runs inside a Java Virtual Machine (VM) to

interoperate or bi-directional interface with application and libraries that would include first library and second library, chapter 1, especially page 1);

the virtual machine interface defines operations performed by the virtual machine implementation-dependent (native code accesses Java VM features by calling JNI functions, page 7 of chapter 2);

a virtual machine interface enables communication between the virtual machine implementation and the first support library and communication between the virtual machine implementation and the second support library (the Java Native Interface (JNI) is a native programming interface. It allows Java code that runs inside a Java Virtual Machine (VM) to interoperate or bi-directional interface with application and libraries that would include first library and second library, chapter 1, especially page 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Folliot and Javasoft because the JNI provides no restrictions on the implementation of the Java virtual machine that allows Java vendors can add support for the JNI without affecting other parts of the virtual machine (Javasoft, page 1).

10. **As to claim 17**, Folliot teaches a runtime environment, embodied in at least one tangible computer-readable medium (memory, page 177) and, suitable for use in conjunction with one virtual machine implementation (VVM, section 1 page 176 –

Art Unit: 2194

section 2) and a plurality of support libraries (VMlets, section 2 pages 177 - 178), the runtime environment comprising:

a virtual machine implementation, embodied in at least one computer-readable medium (memory, page 177) and executable utilizing at least one processing unit (virtual processor, page 178);

a plurality of support libraries, comprising a first support library (VMlets would including a first VMlet, section 2 pages 177 – 178) including library functions with at least some of the libraries functions able to call virtual machine implementation-dependent functions that perform virtual machine implementation-dependent operation (each VMlet include five things: mapping from the VM bytecodes to the VVM's execution mechanism and the VM..... VP instructions, section 2 pages 177 - 178);

wherein the runtime environment is suitable for use in conjunction with a second support library from the plurality of support libraries, capable of replacing the first support library, that has library functions with at least some of the library functions able to call virtual machine implementation-dependent functions that perform virtual machine implementation-dependent operations, the second support library able to conform to the virtual machine interface (unload and load the VMlets, section 1 page 176 paragraph 4 - section 2).

Folliot does not explicitly teach the step of:

a virtual machine interface defining a number of operations performed by the virtual machine implementation;

a virtual machine interface that facilitates communications between a first support library and the virtual machine implementation, the virtual machine interface defining virtual machine dependent operations performed by the virtual machine implementation.

JavaSoft teaches:

a virtual machine interface defining a number of operations performed by the virtual machine implementation (the Java Native Interface (JNI) is a native programming interface. It allows Java code that runs inside a Java Virtual Machine (VM) to interoperate or bi-directional interface with application and libraries that would include first library and second library, chapter 1, especially page 1);

a virtual machine interface that facilitates communications between a first support library and the virtual machine implementation, the virtual machine interface defining virtual machine dependent operations performed by the virtual machine implementation (the Java Native Interface (JNI) is a native programming interface. It allows Java code that runs inside a Java Virtual Machine (VM) to interoperate or bi-directional interface with application and libraries that would include first library and second library, chapter 1, especially page 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Folliot and JavaSoft because the JNI provides no restrictions on the implementation of the Java virtual machine that allows Java vendors can add support for the JNI without affecting other parts of the virtual machine (JavaSoft, page 1).

Response to Arguments

11. Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection.

12. Applicant argued that Folliot does not teach a virtual machine interface that supports communication between one of the VMlets, capable of replacing another VMlet, and the VVM without modification of the virtual machine interface or the VVM. Loading and unloading of VMlets is not replacing as each VMlet performs a different function, i.e. translating the specific bytecode language associated with that VMlet into VVM instructions. Thus, the VMlets are not capable of replacing other VMlets (page 7).

In response, when one VMlet is unloaded and another VMlet is loaded, the loaded VMlet replaces the unloaded VMlet. This operation does not need any modification to the virtual virtual machine. According to Folliot, each VMlet include five elements: mappings from the VM bytecodes to the VVM's execution mechanism and from the VM objects to the VVM's object format, implementations for the VM's primitives, a loader for the VM's applications, and the type checking specification and dynamic security rules for a given language (p. 178). All VMlets include the same five elements; however, the five elements in each VMlet are configured for a specific language that the VMlet supports. At a higher level, each VMlet performs the functionality of translating applications written in a specific language into internal representations of the VVM's execution mechanism (p. 177). For example, a first application written in language_A would require VMlet_A and a second application

written in language_B would require VMlet_B. When execution of the first application is completed and execution of the second application begins, VMlet_A is unloaded and replaced with VMlet_B.

Although no specifically disclosed, it is noted that VMlets in Folliot must communicate with the virtual virtual machine in order for the VMlet to allow the translation of an application into the internal representations used by the VVM's execution mechanism (p. 177 of Folliot). For example, bytecodes from the application is sent to the VMlet, which includes mapping from the VM bytecodes to the VVM's execution mechanism (p. 178 of Folliot), and the translated data is returned to the VVM for execution. In addition, Javasoft teaches Java Native Interface that allows Java code that runs inside a Java Virtual Machine (VM) to interoperate or bi-directional interface with applications and libraries (chapter 1 page 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of Folliot and Javasoft because the JNI provides no restrictions on the implementation of the Java virtual machine that allows Java vendors can add support for the JNI without affecting other parts of the virtual machine (Javasoft, page 1).

Conclusion

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to PHUONG N. HOANG whose telephone number is

Art Unit: 2194

(571)272-3763. The examiner can normally be reached on Monday - Friday 9:00 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyunh S. Sough can be reached on 571-272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/P. N. H./
Examiner, Art Unit 2194

/Li B. Zhen/
Primary Examiner, Art Unit 2194